DTIC FILE COPY

Working Paper No. 359

Contract N00014-85-K-0143

REDUCING CONVEX PROGRAMS WITH TREE CONSTRAINTS

by

Christopher S. Tang

11 May 1988

WESTERN MANAGEMENT SCIENCE INSTITUTE
University of California, Los Angeles

May 11, 1988

Reducing Convex Programs with Tree Constraints

Christopher S. Tang

Anderson Graduate School of Management

University of California, Los Angeles

California, U.S.A.

Abstract

This paper describes a class of convex programs with tree constraints that has applications in production planning, capacity expansion, and other related areas. A reduction procedure is presented for solving this class of convex programs with N variables. This reduction procedure determines an optimal solution to the convex problem by solving at most N simple convex subproblems. Hence, this reduction procedure is an efficient approach for solving large scale convex programs of this sort. (KR)

1. Introduction

Consider a directed tree T that consists of a set of nodes N and a set of directed arcs E, where E is a subset of NxN. For each arc $(j,i) \in E$ that emanates from node j and terminates at node i, node j is an immediate predecessor of node i and node i is an immediate successor of node j. Hence, the root node of T has no immediate successor and each terminal node of T has no immediate predecessor. For each node i that is not the root node nor a terminal node, node i has exactly one immediate successor S(i) and a set of immediate predecessors P(i) such that $(j,i) \in E$ for each $j \in P(i)$. Each node i is associated with a decision variable x_i. Each decision variable x_i is required to satisfy a set of *tree constraints* such that $x_i \le x_j$ for each $j \in P(i)$. Let x be a $|N|$x1 vector whose i-th entry equals x_i. Then, the convex program with tree constraints can be formulated as :

$$\text{(P)} \qquad \text{Min} \quad f(x)$$

$$\text{(1)} \qquad \text{st} \quad x_i \le x_j \qquad \text{for } j \in P(i) \text{ and for each } i \in N$$

$$\text{(2)} \qquad l_i \le x_i \le u_i \qquad \text{for each } i \in N,$$

where f is assumed to be continuous, differentiable and strictly convex over $\Re^{|N|}$. Also, l_i and u_i are the lower and upper bounds on each decision variable x_i; respectively. A feasible solution to problem (P) is a set of x_i's that satisfies the "box" constraints (2) and the tree constraints (1). For each node i, let A(i) denote the set of predecessors of node i. Then it is easy to check that problem (P) is feasible if and only if $l_i \le u_j$ for each $j \in A(i)$ and for each $i \in N$. For this reason, we assume that $l_i \le u_j$ for each $j \in A(i)$. An optimal solution to problem (P) is a feasible solution with minimum objective value.

Suppose we drop the tree constraints (1) in problem (P). Then problem is reduced to the following simple convex subproblem:

$$\text{(S)} \qquad \text{Min} \quad f(x)$$
$$\text{st} \quad l_i \le x_i \le u_i \qquad \text{for each } i \in N.$$

Notice that Problem (S) is a simple convex program with the "box" constraints and it can be easily solved by considering the first order condition (c.f. Luenberger (1984)). The simplicity of the subproblem (S) motivates a reduction scheme, which is described as follows.

Suppose we solve simple convex subproblem (S). Let y denote an optimal solution of problem (S). If y satisfies all the tree constraints, then y is also an optimal solution to problem (P); otherwise, we identify a pair of nodes i and j, and we show that problem (P) will have an optimal solution x^* such that $x^*_j = x^*_i$. In the latter case, since x_i and x_j have the same optimal value, we can replace x_j by x_i in the objective function, and remove x_j from the set of decision variables. Also, we update the remaining tree by setting $S(k) = i$ for each node $k \in P(j)$. Hence, the number of decision variables is reduced by 1 after each repetition of this cycle, and hence, a solution to problem (P) becomes trivial after at most $|N|$ repetition of this cycle. Notice that we only solve the "updated" subproblem (S) in each cycle. Hence, the reduction scheme solves problem (P) by solving at most $|N|$ simple convex subproblems (S). Hence, problem (P) can be quickly solved.

To our knowledge, problem (P) is studied for the first time. Geoffrion (1967) examined a method that solves general convex programs with linear constraints by solving a finite sequence of subproblems. This paper presents a reduction scheme that solves problem (P), a special class of convex programs, by solving at most $|N|$ simple convex subproblems.

This paper is organized as follows. Section 2 discusses the applications of problem (P). Section 3 presents a reduction procedure that solves problem (P) by solving at most $|N|$ simple convex subproblems. Section 4 concludes this paper.

2. Applications of problem (P)

Consider a special case of problem (P) in which the objective function f is separable. Schwarz and Schrage (1975) discussed the applications of problem (P) in production planning. In addition, Willaims (1982) showed the applications of problem (P) in project selection. We omit the details.

When f(x) is not separable, problem (P) has different applications in production planning, capacity planning, and quality improvement. These applications are now described, below.

2.1 Dynamic lot sizes

We schedule the production over a finite planning horizon with N periods. At each period i, let D_i denote the cumulative demand from period 1 to period i and let x_i be the decision variable that represents the cumulative production volume from period 1 to period i. (For notational convenience, let $x_0 = 0$.) Thus, the "actual" production volume in period i is $(x_i - x_{i-1})$, for $i = 1, \ldots, N$. Assume that the demand is drawn at the end of each period. Then $(x_i - D_i)$ represents the inventory level (backorder level) at the end of period i when $(x_i - D_i)$ is positive (negative). The inventory holding cost/backorder cost of carrying z_i units at period i is $h_i(z_i)$. The production cost for producing z_i units at period i is $p_i(z_i)$. Hence, the total cost incurred at period i can be rewritten as $p_i(x_i - x_{i-1}) + h_i(x_i - D_i)$. To determine an optimal production schedule that minimizes the total cost incurred over the planning horizon, we formulate the following optimization problem:

$$\text{(T)} \qquad \text{Min} \quad f(x) = \Sigma_i \; \{p_i(x_i - x_{i-1}) + h_i(x_i - D_i)\}$$

$$\text{(3)} \qquad \text{st} \quad 0 = x_0 \leq x_1 \leq \ldots x_{i-1} \leq x_i \; \ldots \leq x_{N+1} = D_N .$$

Note that constraints (3) ensures that the cumulative production volume is non-decreasing over time and it guarantees that the total production volume meets the total demand over the planning horizon.

Wagner and Whitin (1957) developed the celebrated dynamic lot sizing model, and studied a different formulation of problem (T) by assuming that the cost functions $p_i(.)$ and $h_i(.)$ are concave. Zangwill (1966) extended this model to the case of backlogging.

In the case when all the cost functions are convex, Veinott (1966) suggested an interesting approach that searches negative cycles over a network. His approach is not efficient for solving large problem. This observation motivates us to develop an efficient approach to solve problem (T). Since problem (T) is a special case of problem (P), it suffices to develop an efficient approach for solving problem (P).

2.2 Capacity planning

It is noteworthy to mention that the dynamic lot-sizing model is related to capacity planning models (c.f. Luss(1982)). To see that, equate the cumulative production volume x_i with capacity level at period i, equate the production cost $p_i(x_i - x_{i-1})$ with the capacity expansion cost. Also, equate the cumulative demand D_i with the demand required at period i, equate the inventory holding/backorder cost $h_i(x_i - D_i)$ with the cost of carrying excessive capacity/the penalty charge of operating below the demand. Hence, the capacity planning problem can be formulated as problem (T). Because problem (T) is a special case of problem (P), and hence, problem (P) could also be applied to capacity planning as well as production planning.

2.3 Quality improvement

Suppose a manufacturer has a choice to select the quality level for each of the N periods. Let q_{i-1} be the quality level of the production process during period i-1. Then the decision maker selects a quality level q_i at the beginning of period i. Suppose that the management strives for good quality and desires to improve the quality levels over time; i.e., $q_{i-1} \leq q_i$ for each i. The cost associated with quality improvement during period i is $I_i(q_i - q_{i-1})$, where $I_i(.)$ is assumed to be an increasing and convex function. For any quality level q_i, the demand function for period i is $D_i(q_i)$, where the demand function $D_i(.)$ is assumed to be increasing and concave. Thus, the demand function $D_i(q_i)$ is rather general and it reflects diminishing returns in demand over quality. Let p_i be the net profit per unit sold during time period i, where p_i = selling price at period i - material cost at period i. Hence, the net profit generated in period i is $p_i D_i(q_i)$. There is a cost $C_i(q_i)$ for maintaining the quality level at q_i throughout the time period i. The cost $C_i(q_i)$ consists of operating cost, machine cost, ...etc. We assume that $C_i(q_i)$ is convex and increasing in q_i. Hence, the optimization problem that determines optimal quality levels over time can be formulated as the following convex program:

(U) Min $f(q)$

(4) st $q_{i-1} \leq q_i$ for $i = 1, \ldots, N$,

(5) $q_N \leq 1$,

where $f(q) = \Sigma_i \{I_i(q_i - q_{i-1}) + C_i(q_i) - p_i D_i(q_i)\}$ and q_0 represents the initial quality level. Since $I_i(.)$ and $C_i(.)$ are convex and since $D_i(.)$ is concave, f(q) is convex. This implies that problem (U) is a special case of problem (P), which indicates that problem (P) has application in quality improvement.

3. Convex program with tree constraints (P)

This section presents a reduction procedure for solving problem (P). First, notice that any directed tree T is acyclic, and hence, it is possible to label all the nodes such that $n > m$ for each $n \in P(m)$. Second, f is strictly convex, and therefore, the optimal solution to subproblem (S), denoted by y, is unique. If y satisfies all the tree constraints, then y is also an optimal solution to problem (P). If there exists some nodes such that $y_m > y_n$, where $n \in P(m)$, then we can designate a node i such that i is the largest node with at least one node n, where $n \in P(i)$ and $y_i > y_n$. Formally,

(6) $\quad i = \max \{ m \in N: \exists\, n \in P(m),\ y_m > y_n \}.$

Let $z = (z_1, \ldots, z_N)$ be the optimal solution to problem (P). Clearly, z satisfies the tree constraints while y does not, and hence, $z \neq y$ and $f(y) < f(z)$. The following proposition relates the optimal solution z to node i and to y, where y is the optimal solution to the subproblem (S).

Proposition 1 For any node $m \in N$, z_m satisfies all of the following conditions:

(7) $\quad$ if $z_m < z_n$ for a node $n \in P(m)$, then $z_n \leq y_n$;

(8) $\quad$ if $z_m < z_n$ for each node $n \in P(m)$, then $z_m \geq y_m$; and moreover,

(9) $\quad$ if $m = i$, then $z_n \geq y_n$ for each node $n \in P(m)$.

Proof: We shall prove Proposition 1 by contradiction. Suppose that the optimal solution violates at least one of the conditions. Then it is sufficient to consider the following cases.

(Case a) Suppose that z violates condition (7). Then there is a node $n \in P(m)$ such that $z_m < z_n$ but $z_n > y_n$. In that case, there must exist a δ such that $\delta > 0$, $z_m \leq z_n - \delta$ and $z_n - \delta \geq y_n$. In this case, let z* be a variant of the optimal solution z, where $z^*_s = z_s$ for $s \neq n$, and $z^*_n = z_n - \quad \leq y_n$. Clearly, z^* can be expressed as a convex combination of z and y; i.e., there exists a set of α_n such that $0 \leq \alpha_n \leq 1$, and that $z^*_n = \alpha_n z_n + (1-\alpha_n) y_n$ for $n = 1, \ldots, |N|$. Also, it is easy to check that z^* is a feasible solution to Problem (P). Combine these two observations with the fact that $f(y) < f(z)$, and that f is strictly convex, we can conclude that $f(z^*) < f(z)$, which contradicts that z is an optimal solution. Hence, this case cannot happen.

(Case b) Suppose that z violates condition (8). Then $z_m < z_n$ for each node n $\in P(m)$, but $z_m < y_m$. In this case, there must exist a δ such that $\delta > 0$, $z_m + \delta \leq z_n$ for each node $n \in P(m)$ and $z_m + \delta \leq y_m$. Let z^* be a variant of solution z, where $z^*_s = z_s$ for all $s \neq m$ and $z^*_m = z_m + \delta \leq y_m$. By applying the same argument used in (a), we can conclude that $f(z^*) < f(z)$, which contradicts that z is an optimal solution. Hence, this case cannot happen.

(Case c) Suppose that z violates condition (9). Then $m = i$ and there exists a node $n \in P(m)$ such that $z_n < y_n$. Let J denote the subset of A(i) whose optimal value equals z_n. Hence, $J = \{ s \in A(i): z_s = z_n \}$. Notice that $n \in J$ and that for any node $s \in J$,

(10) $$z_s = z_n < z_k \text{ for } k \in P(s) \backslash J.$$

It follows frcm (6) and the fact that $s > i$ for each $s \in J$, where $J \subset A(i)$, it is easy to see that $y_n \leq y_s$. This implies that

(11) $$z_s = z_n < y_n \leq y_s \qquad \text{for each } s \in J.$$

It follows from (10) and (11), there must exist a δ such that $\delta > 0$, and that

$$z_s + \delta \leq y_s \text{ for each } s \in J, \text{ and}$$

$$z_s + \delta \leq z_k \text{ for each } s \in J \text{ and for each } k \in P(s) \backslash J.$$

Let z^* be a variant of solution z, where $z^*_s = z_s$ for all $s \notin J$ and $z^*_s = z_s + \delta$ for each $s \in J$. For each node $s \in J$, it is easy to see that

$$z^*_s = z_s + \delta > z_s \geq z_k = z^*_k \qquad \text{for } k \notin J \text{ and } k = S(s),$$

$$z^*_s = z_s + \delta = z_k + \delta = z^*_k \qquad \text{for } k \in J \text{ and } k = S(s),$$

$$z^*_s = z_s + \delta \leq z_k = z^*_k \qquad \text{for each } k \in P(s)\backslash J,$$

$$z^*_s = z_s + \delta = z_k + \delta = z^*_k \qquad \text{for each } k \in P(s)\cap J,$$

$$z^*_s = z_s + \delta \leq y_s \leq u_s \qquad \text{for each } s \in J, \text{ and}$$

$$z^*_s = z_s + \delta \geq z_s \geq l_s \qquad \text{for each } s \in J.$$

These observations imply that z^* is also a feasible solution. By applying the same argument used in (a), we can conclude that $f(z^*) < f(z)$, which contradicts that z is an optimal solution. This case cannot happen, which completes the proof. ||

Given node i, as defined in equation (6), we designate a node j such that y_j has the smallest value among all predecessors in P(i). More formally,

(12) $\qquad j = \text{argmin}\{ y_n : n \in P(i)\}$

It follows from the definition of node i in (6), there must exist a node n such that $n \in P(i)$ and $y_i > y_n$. Combine this observation with the definition of j in (12), we can conclude

(13) $\qquad y_i > y_n \geq y_j .$

The conditions (7), (8), and (9) stated in Proposition 1 and (13) enable us to establish a special relationship between z_i and z_j that is considered in the following Proposition.

Proposition 2 The optimal solution z has this property: $z_j = z_i$.

Proof: It follows from the assumption on the bounds, as stated in section 1, $l_i \leq u_n$ for each $n \in A(i)$. Because $j \in P(i) \subset A(i)$ and $l_i \leq u_n$ for $n \in A(i)$,

(14) $\qquad l_i \leq u_j$.

It follows from (13), the fact that $u_i \geq y_i$, and the fact that $y_j \geq l_j$, we have

(15) $\qquad l_j \leq y_j < y_i \leq u_i$.

Combine (14), (15) and the fact that $l_i \leq u_i$ and the fact that $l_j \leq u_j$, we conclude that

(16) $\qquad \max\{l_i, l_j\} \leq \min\{u_i, u_j\}$.

Hence, there must exist a feasible solution x to problem (P) such that $l_i \leq x_i \leq u_i$, $l_j \leq x_j \leq u_j$, and $x_j = x_i$. It remains to show that the optimal solution z has $z_j = z_i$. Suppose that $z_i \neq z_j$. Since z satisfies the tree constraints and since $j \in P(i)$,

(17) $\qquad z_i < z_j$.

In this case, condition (7) implies that

(18) $\qquad z_j \leq y_j$.

In addition, condition (9) implies that

(19) $\qquad y_n \leq z_n$ for each $n \in P(i)$.

It follow from the definition of j in (12),

(20) $\qquad y_j \leq y_n$ for each $n \in P(i)$.

Combining (19), (20), (18) with (17), we have

(21) $\qquad z_n \geq y_n \geq y_j \geq z_j > z_i$ for each $n \in P(i)$.

It follows from (21) and condition (8),

(22) $\qquad z_i \geq y_i$.

Combine (22), (13) and (18), we can conclude that

(23) $\qquad z_i \geq y_i > y_j \geq z_j$.

Hence, $z_i > z_j$, which contradicts (17). This completes the proof. ||

We now apply Proposition 2 to construct a reduction procedure. Suppose we compute y by solving problem (S). If y satisfies all the tree constraints, then y is also an optimal solution to problem (P). If y violates some of the constraints, then we determine a specific node i and a specific node j such that $z_j = z_i$. In the latter case, we reduce the set of decision variables $\mathcal{L}$ by setting $\mathcal{L} = \mathcal{L} \setminus j$, update the objective function by replacing x_j by x_i, replace l_i by Max (l_i , l_j), and replace u_i by Min (u_i , u_j). We update the remaining tree by setting S(k) = i for each k ∈ P(j). This allows us to start the next cycle. Hence, an optimal solution to problem (P) can be found within |N| cycles. Let $\mathcal{L}$ be the current set of decision variables. Table I, below, presents the reduction procedure that solves problem (P) by solving at most |N| simple convex subproblems.

Reduction Procedure

Step 1: Set $\mathcal{L}$ = N.

Step 2: [Optimality] Compute y_k for each $k \in \mathcal{L}$. If $y_k \leq y_l$ for $l \in P(k)$ and for each k, set $z_k = y_k$ for each $k \in \mathcal{L}$ and STOP; otherwise, continue.

Step 3: [Reduction] Compute i and j. Replace x_j by x_i in the objective function, set l_i = Max {l_i , l_j}, set u_i = Min { u_i , u_j}, and replace $\mathcal{L}$ by $\mathcal{L} \setminus j$. Also, set S(k) = i for each k ∈ P(j). GOTO Step 2.

Table I

At each iteration, the reduction procedure either stops and finds an optimal solution in Step 2 or it reduces the number of decision variables by one in Step 3. Thus, the reduction procedure stops within |N| iterations. We solve only one convex subproblem (S) in Step 2. Hence, the reduction procedure solves problem (P) by solving at most |N| simple convex subproblems.

4. Concluding Remarks

Throughout this paper, we assumed that f is continuous and differentiable. This assumption guarantees that the first order condition can be easily obtained, and hence, problem (S) can be quickly solved. In addition, we assumed that f is strictly convex. This assumption provides computational advantages for computing the unique optimal solutions of problem (P) and of its subproblem (S). It can be shown, however, that the reduction procedure can be adapted to the case when f is not strictly convex. In that case, the "adapted" reduction procedure would not be efficient because of two reasons. First, when f is not strictly convex, the set of optimal solutions to problem (S) is a convex set C, say, which may not have a simple characterization. Second, given the set C, it is not easy to check whether there exists an optimal solution $y \in C$ such that y satisfies all the tree constraints.

In sum, we presented a class of separable convex programs with tree constraints that has applications in production planning, capacity planning, quality improvement and other related areas. By exploring some special properties of the optimal solution to problem (P), we developed a reduction procedure that solves problem (P) by solving at most |N| simple convex subproblems, and hence, problem (P) can be quickly solved. Therefore, this reduction procedure is efficient for solving large convex programs of this sort. As a future research direction, we shall examine the case when the constraints are more general than the tree constraints; for instance, network constraints.

References

Geoffrion, A., (1967), "Reducing Concave Programs with some Linear Constraints," *SIAM J. Appl. Math.* Vol. 15, pp. 653-664.

Luenberger, D.G., (1984), *Linear and Nonlinear Programming*, Second edition, Addison Wesley, Massachusetts.

Luss, H., (1982), "Operations Research and Capacity Expansion Problems: A Survey," *Operations Research*, 30, pp. 907-947.

Schwarz, L. B., and L. Schrage, (1975), "Optimal and System Myopic Policies Multi-Echlon Production/Inventory Assembly Systems," *Management Science*, Vol. 21, pp. 1285-1294.

Veinott, A. F., Jr. (1964), "Production Planning with Convex Costs: A Parametric Study," *Management Science*, 10, pp. 441-460.

Wagner, H.M., and T. Whitin, (1957), "Dynamic Problems in the Theory of the Firm," *Theory of Inventory Management*, 2nd ed., T. Whitin, ed., Princeton University Press, Princeton, N.J., App. 6.

Williams, H.P., (1982), "Models with Network Duals," *Journal of Operational Research Society*, Vol. 33, pp. 161-169.

Zangwill, W.I., (1966), "A Deterministic Multi-period Production Scheduling Model with Backlogging," *Management Science*, 13, pp. 105-119.